# VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data

Hank Childs∗,§ , Eric Brugger† , Brad Whitlock† , Jeremy Meredith‡ , Sean Ahern‡ , Kathleen Bonnell† , Mark Miller† ,
Gunther H. Weber∗ , Cyrus Harrison† , David Pugmire‡ , Thomas Fogal¶ , Christoph Garth§ , Allen Sanderson¶ , E. Wes Bethel∗ ,
Marc Durant▽, David Camp∗,§ , Jean M. Favre€ , Oliver Rubel∗ , Paul Navratil△ , Matthew Wheelerα , Paul Selbyα , and Fabien Vivodtzev±

∗ Lawrence Berkeley National Laboratory, † Lawrence Livermore National Laboratory, ‡ Oak Ridge National Laboratory, § University of California at Davis,
¶ University of Utah, ▽Tech-X Corporation, € Swiss National Supercomputing Center, △Texas Advanced Computing Center,
α Atomic Weapons Establishment, ± French Atomic Energy Commission, CEA/CESTA
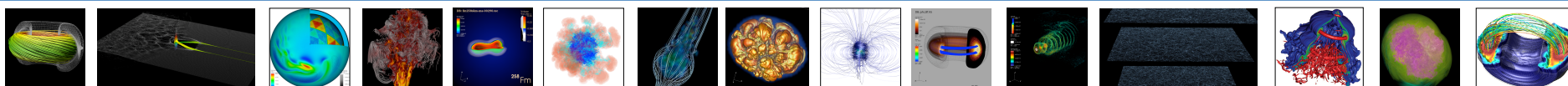
Project Goals

VisIt is built around three primary goals:
1) Enable data understanding. This is done by supporting five broad use cases: data exploration, visual debugging, comparative analysis, quantitative analysis, and communication.
2) Support very large data sets, including the large data sets being generated on today's ASCR machines
3) Provide a robust and usable product for end users.
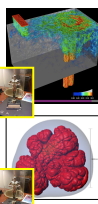
Project Design

VisIt follows a client-server design. The client is run on the user's local desktop machine and performs tasks requiring interactivity. The server runs where the user's data resides, typically supercomputers, and processes the data in parallel. A major strength of VisIt's interface is interoperability: its 115 file format readers, 60 operators for data manipulation, 20 rendering methods, 190 methods for creating derived data, and 90 queries for extracting quantitative and debugging information all work together to create a powerful environment.
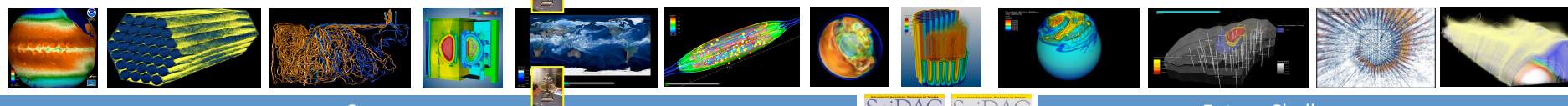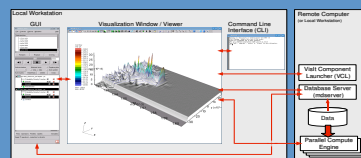
Successes

We describe four types of success achieved with VisIt:

1) User successes: Metrics on user success are difficult. Downloads are one metric of user interest; VisIt has been downloaded over 200,000 times. In terms of SciDAC usage, over a dozen SciDAC code groups use VisIt as their workhorse visualization tool and notable images from those groups are throughout this poster. In terms of institutions, LBNL and ORNL do not keep user statistics, but their visualization teams view VisIt as their primary tool. ANL also makes heavy use of the project. Further, LLNL has 300 regular VisIt users, AWE has 100, and CEA/CESTA has 50.

2) Scalability successes: In 2009, a pair of studies were run to demonstrate VisIt's capabilities for scalability and large data. In the first study, VisIt's infrastructure and some of its key visualization algorithms were demonstrated to support weak scaling. This demonstration led to VisIt being selected as a "Joule code," a formal certification process by the US Office of Management and Budget to ensure that programs running on high end supercomputers are capable of using the machine efficiently. In the second study, VisIt was scaled up to tens of thousands of cores and used to visualize data sets with trillions of cells per time slice. This study found VisIt itself to perform quite well, although overall performance was limited by the supercomputer's I/O bandwidth.

3) A repository for large data algorithms: Every algorithm in VisIt works in parallel and operates on large data sets. Some of these algorithms required novel implementation that were published in research literature, most notably including particle advection, volume rendering, and connected component identification. Further, VisIt's large data processing is managed by its innovative contract-based execution system.

4) Supercomputing research performed with VisIt: VisIt has been used for evaluating the benefits of evolving hardware trends, such as GPUs, hybrid parallelism, and solid state drives.

Future Challenges

VisIt will face many challenges in the future:

In the short term, I/O limitations will force I/O to be de-emphasized. The VisIt development team has invested in pertinent techniques, such as multi-resolution processing and in situ, but these techniques will need to be further hardened to support regular production use.

In the longer term, power limits will constrain data movement, forcing much processing to occur in situ on novel architectures, such as GPU accelerators. Unfortunately, VisIt's existing in situ implementation may be mismatched for this many-core future, for two reasons. First, although VisIt can be easily multi-threaded using a pthreads or OpenMP-type approach, this approach may not be able to take advantage of these architectures. The many-core future may require CUDA or OpenCL-type languages; migrating the VisIt code base to this setting would be a substantial undertaking. Second, although VisIt has been demonstrated to work well at high levels of concurrency, some of its algorithms involve large data exchanges. Although these algorithms perform well on current machines, they would violate the data movement constraints on future machines and may need to be re-designed.
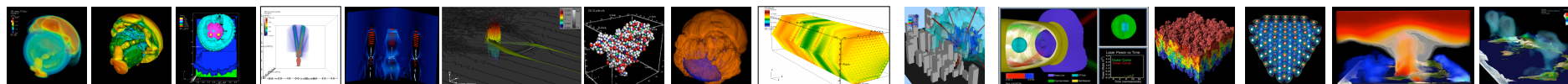
VACET & VisIt

In 2006, VACET was funded to deliver petascale-capable visualization and analysis infrastructure to the SciDAC and INCITE communities. VACET chose VisIt as its primary deployment vehicle and VACET personnel made many changes: they worked with many application groups and extended VisIt to meet their needs, they overhauled several crucial algorithms to perform better on large data, and they made changes so that VisIt could scale to extremely high levels of concurrency and operate on very large data, leading to its selection as a Joule code. VACET successfully got SciDAC end users to adopt VisIt through a "teach them to fish" approach that included heavy outreach, tutorials at the SciDAC conferences, SC10, and more.